A High-Level Policy Description Language for the Network ACL

Jangha Kim[1], Kanghee Lee[1], Sangwook Kim[1],
Jungtaek Seo[2], Eunyoung Lee[2], and Miri Joo[2]

[1] Department of Computer Science, Kyungpook National University,
1370 Sankyuk-dong Buk-gu, Daegu, 702-701, Korea
[2] National Security Research Institute,
161 Gajeong-dong Yuseong-gu, Daejeon, 305-350, Korea
{jhkim, khlee, swkim}@cs.knu.ac.kr, {seojt, eylee, mrjoo}@etri.re.kr

Abstract. Malicious codes and worms comprise the largest portion of the loss caused the security problem in the Internet. Small worms such as the "Blaster" spread quickly through the enormous network. It causes the network to lock down within an hour or so[1]. The situation worsens before it can be monitored and notified by the supervisor. Since the network is not available, it becomes hard to serve a node with an order. It is difficult for most large networks to introduce a consistent monitoring tool and reporting system. It is also more difficult to manage the configuration of network nodes with the matter of policy. We represent abstract language that supports various functions. Functions are in grouping, event, compliance and intermediate forms. This high-level language abstracts the control behavior of the network nodes that have various setting-up methodologies. We will describe the features of the language and give examples of the preliminary implementation on the test-bed.

1 Introduction

The most important process in the management of network security is the protection of the network from automated worms that spread quickly through the Internet. It is for this reason that the abstract policy language and hierarchy security management system are necessary in delivering security information such as the policies among the nodes and domains. The purpose of this study is to design and implement a high-level ACL description language for the large-scale network of the hierarchy security management system. The large-scale network consisting of many hosts and nodes is divided into sub-networks.

The rest of this paper is organized as follows: section 2 describes a specific large-scale network and explains the necessity of high-level security management; section 3 reviews the features of the proposed high-level language; section 4 and 5 explain the environment of implementation and show the results of its comparisons to the other high-level languages; and section 6 presents the conclusion and future applications.

K.-M. Liew et al. (Eds.): PDCAT 2004, LNCS 3320, pp. 748–751, 2004.

2 The Large Network and the Abstraction of Policy

In this paper, the organization that forms itself into a logical hierarchy from highest to the lowest is called the "Domain". The generic spread process of the malicious code transfers harmful traffic from one infected domain to another. To prevent the spread of harmful traffic in the system requires collaboration between the Domains. When harmful traffic is detected, coherent policies have to report within the Domains.

In constructing hierarchal Domains in the large-scale network, each Domain is recursively constructed. The highest Domain, such as the back-born, is the highest network not only logically but also physically. The highest Domains are constructed as 'Nets'. Many researches connected to the collaboration of the highest Domains are currently in progress. To keep in line with the purpose of this paper, we will not take this matter up in detail. Figure 1 represents the Domain constitution of the large-scale network described above.

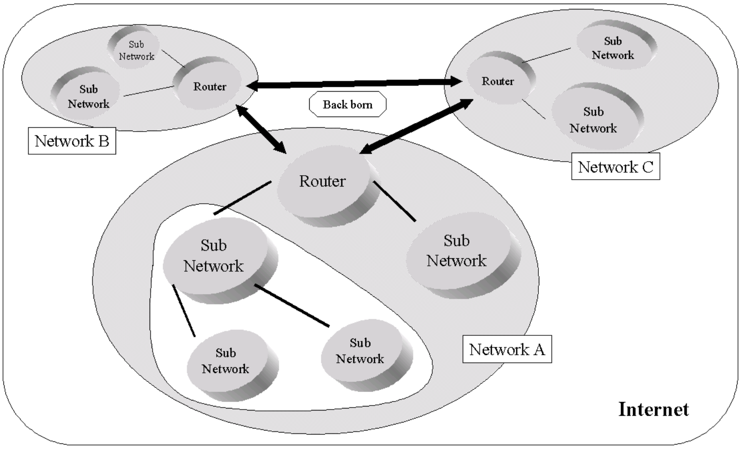


Fig. 1. The constitution of the Domain of large-scale networks

3 Triton Language

The Domain management policies must describe the high-level abstract language used to control the large-scale network in the block. It is necessary to describe the heterogeneous and configurative information of various network nodes. This approach provides the highest Domain with a methodology to manage the large-scale network. A high-level Triton offers various political functions that manage lower network nodes. This mechanism provides five perspectives of a description:

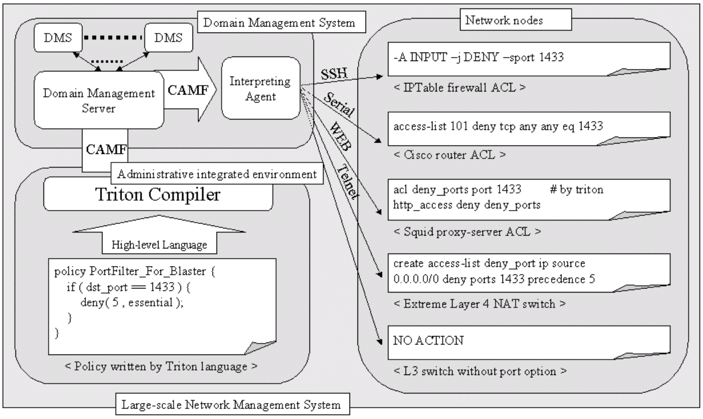
- Abstract description for the configuration of a lower node
- Mechanism of policy compliance
- Grouping
- Event for a polymorphous policy
- Semi-structured communication framework

We will illustrate with a simple example[2].

```
policy SamplePolicy triggered by EVENT_ALERT
{
    Range R1 = [ x:IP | "10.1.1.5" <= x <= "10.1.1.20" ];

    incoming {
        for ( "DomainA", "DomainB", "DomainC" ) {
            if ( src_addr in R1 && dst_port == 8080 ) {
                deny ( 3, essential );
            }
        }
    }
}
```

The other features have to be supported by the compiler, especially those that require mathematical notations in order to process text files. The description language is composed of a group of structure dependent statements. The statement attributes also play an important role in defining a policy. The system that is going to be designed will also include an editor creating the policy. A Triton compiler can process an instantiated policy and build a management system with the structure and features described in the document.


Domain Management System
DMS
DMS
Domain Management Server
CAMF
Interpreting Agent
CAMF
Administrative integrated environment
Triton Compiler
High-level Language
< Policy written by Triton language >
Network nodes
SSH
Serial
WEB
Telnet
< IPTable firewall ACL >
< Cisco router ACL >
< Squid proxy-server ACL >
< Extreme Layer 4 NAT switch >
< L3 switch without port option >
Large-scale Network Management System

```
policy PortFilter_For_Blaster {
   if ( dst_port == 1433 ) {
      deny( 5 , essential );
   }
}
```

```
-A INPUT –j DENY –sport 1433
```

```
access-list 101 deny tcp any any eq 1433
```

```
acl deny_ports port 1433        # by triton
http_access deny deny_ports
```

```
create access-list deny_port ip source
0.0.0.0/0 deny ports 1433 precedence 5
```

```
NO ACTION
```

Fig. 2. An abstract language for the configuration of the lower node

4 Evaluation

This paper compares the Triton language to the other languages representing policies for the evaluation of a proposed language. There are various policies description languages used in managing network security. Most of the languages have individual properties. We looked into the policy description language, with a similar purpose, for the comparison and selection of five network policy languages[3][4].

Table 1. The comparison of police represented languages

Language	Target Architecture	Compliance	Group	Event	Reference	Represent
Policy term	Distributed	Low	Low	Low	Low	Low
PFDL	Centralized	Medium	Low	Medium	Low	Low
RPSL	Centralized	Low	Low	Low	Low	Low
PAX	Centralized	Low	Low	Low	Low	Low
SRL	Distributed	Low	Low	Low	Low	Low
NetSPoC	Centralized	Low	High	Low	Low	Medium
Triton	Distributed	High	Medium	Medium	Medium	High

5 Conclusion

The policy manager has summarized and abstracted the information on large-scale networks to set up a coherent policy. Additionally, if the Domain server accommodates a policy delivered from the other trustable Domains, the lower Domain then, is improving in security. In the lower Domain, it collects information of harmful traffic and forwards the useful information to the higher Domain. In the higher Domain, it sends useful policies to the lower Domains. In this manner, the high-level ACL description Language and related management systems perform abstracted policy descriptions and various policy services, such as compliance to policy, grouping, an event-based approach and an intermediate form. This gives a tried manager the important function of managing the security of the large-scale network.

References

1. Mohit Lad, Xiaoliang Zhao, Beichuan Zhang, Dan Massey and Lixia Zhang : "An Analysis of BGP Update Burst during Slammer Attack," IWDC, Calcutta, 2003
2. Triton BNF, http://woorisol.knu.ac.kr/lab/content/triton-bnf.txt, 2004
3. G.N.Stone, B.Lundy, and G.G.Xie : "Network Policy Languages: A Survey and a New Approach," IEEE Network, Vol.15, No.1, pp. 10-20, 2001
4. NetSPoc homepage, http://netspoc.berlios.de/, 2004